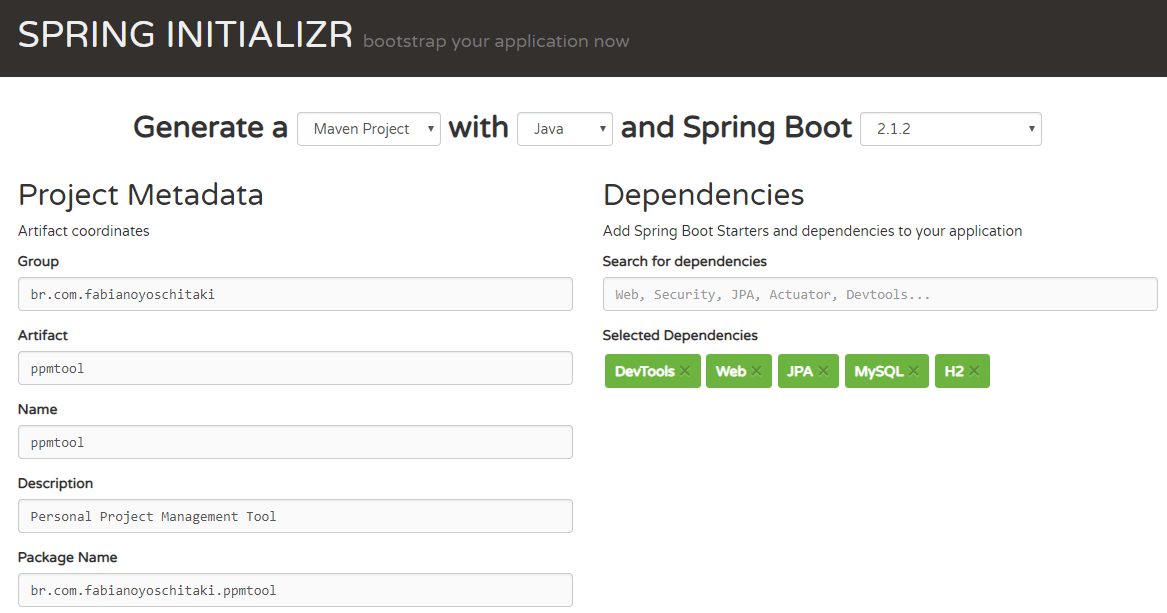
**Final project: https://agileintppmtool.herokuapp.com**

* **Section 2: SPRING BACKEND - BASIC CRUD OPERATIONS – PROJECT**
  + **Folder Structure and Github setup**



Import the Maven project to your IDE.

Add project to github (parent folder containing ppmtool-java). Type git branch to see how many branches you have (only master, which has the stable tool with unit/integration/regression tests).

Create branch **git branch branch0** and then **git checkout branch0**

Make any changes, then **git add .** > **git commit -m “changes”** >

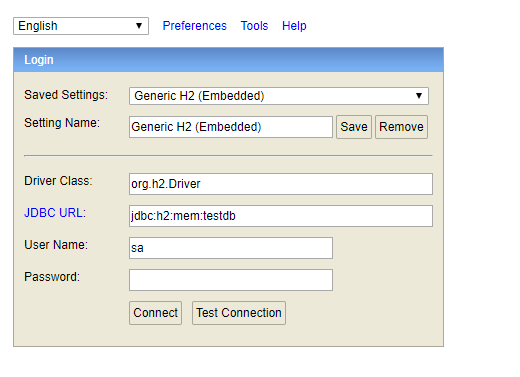
git push --set-upstream origin branch0

Now to add it to the master branch: **git checkout master > git merge branch0.**

* + **Project Object & Project Repository-branch1**

Created Project.java Entity, run Spring Boot App and go to H2 console: <http://localhost:8080/h2-console>

Change JDBC Url to **jdbc:h2:mem:testdb**

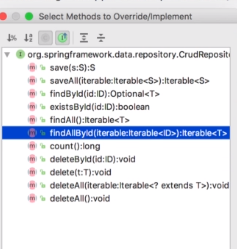


Created the interface ProjectRepository:

***@Repository***

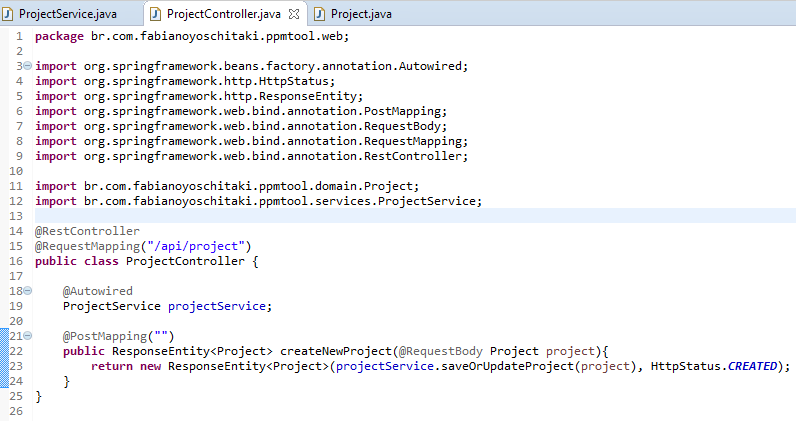
**public** **interface** ProjectRepository **extends** CrudRepository<Project, Long>{

}

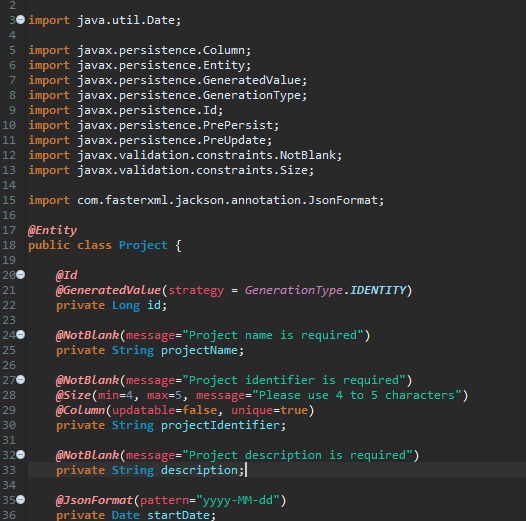


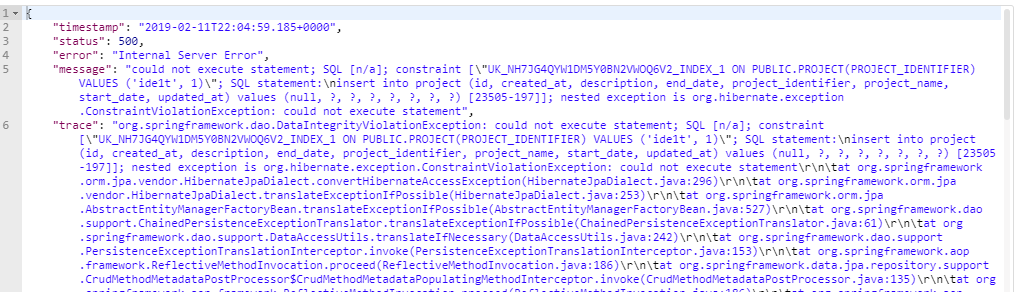
* + **Project Service & Project Controller | Create first project - branch2**





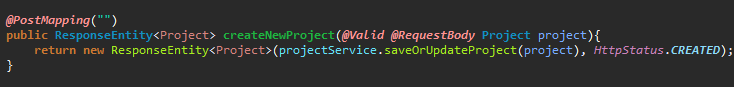
* + **Set up Project Object Validation – branch3**



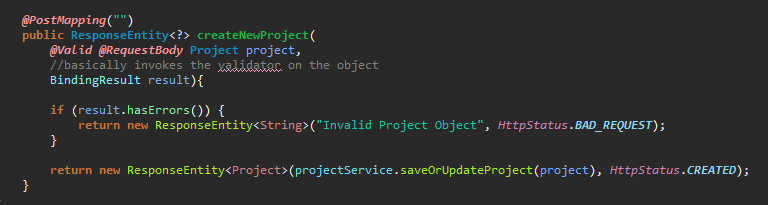


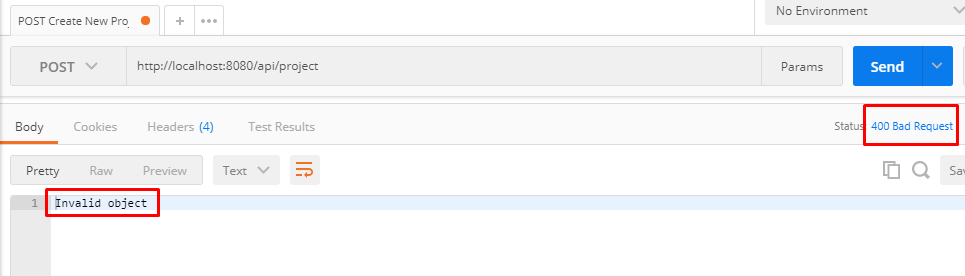
* + **Project Object Validation part1 – branch4**

Adding @Valid makes the error looks better.



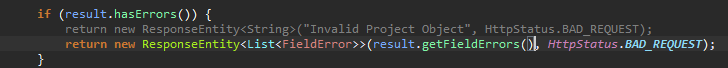






* + **Project Object Validation part2 – branch4**

Let’s set a JSON response error if validation has errors. We can extract from BindingResult the list of errors:





Let’s return only field and defaultMessage:

